

# Metropolising forward particle filtering backward sampling and Rao-Blackwellisation of Metropolised particle smoothers

Jimmy Olsson      Tobias Rydén

November 10, 2010

## Abstract

Smoothing in state-space models amounts to computing the conditional distribution of the latent state trajectory, given observations, or expectations of functionals of the state trajectory with respect to this distributions. For models that are not linear Gaussian or possess finite state space, smoothing distributions are in general infeasible to compute as they involve integrals over a space of dimensionality at least equal to the number of observations. Recent years have seen an increased interest in Monte Carlo-based methods for smoothing, often involving particle filters. One such method is to approximate filter distributions with a particle filter, and then to simulate backwards on the trellis of particles using a backward kernel. We show that by supplementing this procedure with a Metropolis-Hastings step deciding whether to accept a proposed trajectory or not, one obtains a Markov chain Monte Carlo scheme whose stationary distribution is the exact smoothing distribution. We also show that in this procedure, backward sampling can be replaced by backward smoothing, which effectively means averaging over all possible trajectories. In an example we compare these approaches to a similar one recently proposed by Andrieu, Doucet and Holenstein, and show that the new methods can be more efficient in terms of precision (inverse variance) per computation time.

## 1 Introduction

The topic of the present paper is computation of smoothed expectations of functionals of the state process in state-space models, i.e. conditional expectations of such functionals given data. To make this discussion more

precise, let  $(\mathbf{X}, \mathbf{Y})$  be a *state-space model*, where  $\mathbf{Y} = (Y_k)_{k=0}^n$  is the observable (output) process and  $\mathbf{X} = (X_k)_{k=0}^n$  is the latent (unobserved) Markov chain. The relation between the two is such that given  $\mathbf{X}$ , the  $Y_k$ 's are conditionally independent with the conditional distribution of a particular  $Y_k$  depending on the corresponding  $X_k$  only. If the state space of  $\mathbf{X}$  is finite we use the term *hidden Markov model* (HMM). Our interest thus lies in computing conditional expectations of the form  $\mathbb{E}[h(X_{0:n}) | y_{0:n}]$  for a real-valued functional  $h$  of one, several, or all of the latent variables. Here  $X_{0:k} = (X_0, X_1, \dots, X_k)$  etc.; this form will be our generic notation for vectors.

Conditional expectations as above are often interesting and relevant in their own respect, with e.g.  $\mathbb{E}[X_k | Y_{0:n}]$ ,  $\mathbb{E}[X_k^2 | Y_{0:n}]$  and  $\mathbb{P}(X_k \geq x | Y_{0:n}) = \mathbb{E}[\mathbb{1}(X_k \geq x) | Y_{0:n}]$ , where  $\mathbb{1}(\cdot)$  denotes some indicator function, providing useful inferential summaries of the latent states. Another very common use of such expectations is however for inference on model parameters through the EM algorithm. Indeed, assume that the distribution of  $(\mathbf{X}, \mathbf{Y})$  depends on some model parameter (vector)  $\theta$ . Then in the E-step of the EM algorithm, it is typical that conditional expectations with functionals like  $h(x_{0:n}) = \sum_{k=0}^n x_k$ ,  $h(x_{0:n}) = \sum_{k=0}^n x_k^2$ ,  $h(x_{0:n}) = \sum_{k=1}^n x_{k-1}x_k$  etc. appear, in particular if the joint distribution of  $(\mathbf{X}, \mathbf{Y})$  belongs to an exponential family of distributions. For HMMs, the functional  $h(x_{0:n}) = \sum_{k=1}^n \mathbb{1}(x_{k-1} = i, x_k = j)$  is used to re-estimate the transition probability from state  $i$  to state  $j$ . Unfortunately, exact numeric computation of the conditional distribution of  $X_k$ , or that of a sequence of  $X$ 's, given  $Y_{0:n}$ , is possible essentially only in two cases. Firstly for HMMs, for which the forward-backward algorithm [4] provides the solution, and secondly for linear Gaussian state-space models, for which conditional distributions are Gaussian and the Kalman smoothing recursions provide their conditional means and (co)variances [e.g. 13, Chapter 4]. For other models, i.e. models with continuous state space and non-linear and/or non-Gaussian dynamics and/or output characteristics, there are no exact numerical methods available and one is confined to using approximations. Traditional approaches include Kalman filtering and smoothing techniques based on linearisation of the system dynamics and output characteristics, such as the extended Kalman filter, but following the impact of Markov chain Monte Carlo (MCMC) methods in general, during the last 10–15 years there has been a dramatic increase in the interest in and use of simulation-based methods to approximate conditional expectations given data. When such methods are used to approximate expectations appearing in the E-step of the EM algorithm, one often talks about *Monte Carlo EM* (MCEM) algorithms.

For a model as above it holds that the conditional distribution of  $\mathbf{X}$  given  $\mathbf{Y}$  is that of a time-varying Markov chain. This fact lies behind the existence of the forward-backward algorithm for HMMs, and is also the cornerstone of any algorithm that simulates  $\mathbf{X}$  conditionally on  $\mathbf{Y}$ . To simulate  $\mathbf{X}$  given  $\mathbf{Y}$  and a fixed set of parameters  $\theta$ , there are essentially two different approaches. The first one, often referred to as *local updating*, is to run an MCMC algorithm that updates one  $X_k$  at the time, given  $\mathbf{Y}$  and  $X_{-k} = (X_j)_{0 \leq j \leq n, j \neq k}$ . Because of the model structure, the only variables appearing in this conditional distribution are  $Y_k$ ,  $X_{k-1}$  (unless  $k = 0$ ) and  $X_{k+1}$  (unless  $k = n$ ). By varying  $k$  one obtains an MCMC algorithm whose stationary distribution is that of  $\mathbf{X}$  given  $\mathbf{Y}$  [see e.g. 18]. The other approach to simulating  $\mathbf{X}$  given  $\mathbf{Y}$  is simply to simulate a full trajectory from the conditional distribution in question. This can be done either by *forward filtering-backward sampling* (FFBS), or by *backward recursion-forward sampling*. These names stem from the two blocks of the forward-backward algorithm for HMMs, replacing either of them by simulation. In this paper we focus on the former approach. After recursively computing the filter, i.e. the conditional distribution of  $X_k$  given  $Y_{0:k}$ , for  $k = 0, 1, \dots, n$ , forward filtering-backward sampling first simulates  $X_n$  from the filter distribution at time  $n$  and then recursively simulates, for  $k = n - 1, n - 2, \dots, 0$ ,  $X_k$  from the conditional distribution of  $X_k$  given  $X_{k+1}$  and  $Y_{0:k}$ .

Comparing the two approaches, the advantage of local updating is its simplicity as it only involves simulation from univariate (conditional) distributions. Its disadvantage is that a significant burn-in period may be required to remove bias, and that mixing can be slow so that many MCMC iterations are required to make sure that sample means approximate the corresponding conditional expectations with required accuracy. For FFBS on the other hand, one must compute the filter distribution. This is easily done for HMMs [e.g. 7, used FFBS for HMMs], but for continuous state spaces the filter distributions are in general not available. The foremost advantage of FFBS is that the simulated replications of  $\mathbf{X} | \mathbf{Y}$  are independent.

To approximate filter distributions in models with continuous state space, a class of methods known as *particle filters*, or *sequential Monte Carlo* (SMC) methods, has received considerable attention during the last 10–15 years; see e.g. [14, 11, 6] for introductions to such methods, and e.g. [2, 15, 20] for surveys and applications.

Particle filters approximate the filter distribution at time  $k$  by a discrete distribution  $\sum_{i=1}^N \omega_k^i \delta_{\xi_k^i}$ , for which the locations  $\xi_k^i$ , the so-called *particles*, and the non-negative weights  $\omega_k^i$  evolve randomly and recursively in time.

Using such an approximation one can thus recursively compute approximations to the filter distributions, and then use them to simulate a state trajectory backwards. The distribution of this trajectory will then only approximately be that of  $\mathbf{X} | \mathbf{Y}$ . The idea to use particle filters for approximate backward sampling first appeared, to the best of our knowledge, in [12]. The paper [8] provides theory (see e.g. Theorem 5 and Corollary 6 therein) that supports the validity of this approach such as consistency results ensuring that, as the number of particles increases, the distribution of a trajectory sampled using the particle filter converges to the true smoothing distribution.

A recent paper, [1], devised a method related to FFBS but that removes bias entirely by adding a Metropolis-Hastings (M-H) step. The approach is described in detail below, but in short it involves running a particle filter and then selecting a state trajectory not by backward sampling, but by sampling one of the particles at the final time-point  $n$  according to its importance weight, and then tracing the history of this particle back to the first time-point. The M-H step is constructed so that the stationary distribution of the sampled trajectory is indeed the distribution of  $\mathbf{X} | \mathbf{Y}$ . A well-known problem of particle filters is however that as the filter recursion proceeds beyond a given time-point  $k$ , after a while only a few of the particles that existed at  $k$  will have survived the step-wise selection process. This implies that the genealogical tree of the particle filter provides a poor approximation to the smoothing distribution at time-points just a bit prior to current time. FFBS does not suffer from this kind of degeneration, and in the present paper we show how an M-H step can be applied to remove bias also from particle FFBS. We also show how the approach can be Rao-Blackwellised, by which we mean that sampling of trajectories is replaced by the corresponding expectation, which is the backward smoothing recursion. As a compromise between single trajectory sampling and smoothing one may also simulate a small number of trajectories from each set of particles. We analyse the various approaches from a variance/cost perspective, and show that backward simulation and smoothing can be notably more efficient than sampling from the genealogical tree.

We started the work on the material presented in this paper when we during the writing of the manuscript [16], on approximate data augmentation MCMC schemes using particle FFBS, became aware of a preprint version of [1]. Later, in the discussion part following the published version of that paper (p. 306–307), we found that Nick Whitley (University of Bristol) had been thinking along similar lines. Therefore we would like to point out some features of our paper that are not found in Whiteley’s comment,

nor in the paper [1] itself. One such feature is that we allow for general auxiliary particle filters in the MCMC sampler, and another one is the multiple trajectory sampling idea that compromises between single trajectory sampling and backward smoothing, as well the variance/cost analysis of this and other approaches.

## 2 Preliminaries

In this section we sharpen the notation and introduce some basic concepts that will be used throughout the paper. We assume that all random variables are defined on a common probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ . The state space of  $\mathbf{X}$  is denoted by  $\mathcal{X}$ , and by  $\mathcal{Y}$  we denote the space in which  $\mathbf{Y}$  takes its values. We suppose that both of these spaces are Polish and write  $\mathcal{B}(\mathcal{X})$  and  $\mathcal{B}(\mathcal{Y})$  respectively for the corresponding Borel  $\sigma$ -fields. The transition kernel and initial distribution of  $\mathbf{X}$  are denoted by  $Q$  and  $\rho$ , respectively, and we assume that the transition kernel  $Q$  admits a density  $q$  w.r.t. some fixed reference measure  $\lambda$  on  $\mathcal{X}$ , in the sense that

$$Q(x, A) = \int \mathbb{1}_A(x') q(x, x') \lambda(dx')$$

for all  $A \in \mathcal{B}(\mathcal{X})$  and  $x \in \mathcal{X}$ . We also assume that the conditional distribution of  $Y_k$  given  $X_k = x$  has a density  $g(x, \cdot)$  (the *emission density*) w.r.t some reference measure  $\nu$ . In most applications  $\mathcal{X}$  and  $\mathcal{Y}$  are products of  $\mathbb{R}$ , and  $\lambda$  and  $\nu$  are Lebesgue measures. Here we have tacitly assumed that neither  $Q$  nor  $g$  depends on time  $k$ , but the extension to time-varying systems is immediate.

We will throughout the paper assume that we are given a fixed record  $y_{0:n}$  of arbitrary but fixed observations and our main target is to produce samples from the joint posterior distribution  $\phi_{r:s|n}(A) \stackrel{\text{def}}{=} \mathbb{P}(X_{r:s} \in A \mid Y_{0:n} = y_{0:n})$ ,  $A \in \mathcal{B}(\mathcal{X})^{\otimes(s-r+1)}$ , of a record of states given the observations. The special cases  $\phi_k \stackrel{\text{def}}{=} \phi_{k:k|k}$  and  $\phi_{0:n|n}$  will be referred to as the *filter* and *joint smoothing* distributions, respectively. Since the model is fully dominated, each joint smoothing distribution  $\phi_{0:k|k}$  has a density (denoted by the same symbol) w.r.t. products of  $\lambda$  and  $\nu$ . This density is proportional to  $\rho(x_0)g_0(x_0) \prod_{\ell=1}^k g_\ell(x_\ell)q(x_{\ell-1}, x_\ell)$  and we denote by  $Z_k$  the normalising constant. Since the observations are fixed we will keep the dependence of any quantity on these implicit and introduce the short-hand notation  $g_k(x) \stackrel{\text{def}}{=} g(x, y_k)$  for  $x \in \mathcal{X}$ .

It is easily shown that the joint smoothing distributions  $(\phi_{0:k|k})_{k=0}^n$  satisfy the well known *forward smoothing recursion*

$$\phi_{0:k|k}(A) = \frac{\iint \mathbb{1}_A(x_{0:k}) g_k(x_k) Q(x_{k-1}, dx_k) \phi_{0:k-1|k-1}(dx_{0:k-1})}{\iint g_k(x_k) Q(x_{k-1}, dx_k) \phi_{0:k-1|k-1}(dx_{0:k-1})}, \quad (2.1)$$

implying the analogous recursion

$$\phi_k(A) = \frac{\iint \mathbb{1}_A(x_k) g_k(x_k) Q(x_{k-1}, dx_k) \phi_{k-1}(dx_{k-1})}{\iint g_k(x_k) Q(x_{k-1}, dx_k) \phi_{k-1}(dx_{k-1})} \quad (2.2)$$

for the filter distributions. Conversely, the joint smoothing distributions may be retrieved from the filter distribution flow using the so-called *backward decomposition* of the smoothing measure. Indeed, let, for  $A \in \mathcal{B}(\mathcal{X})$ ,

$$\overleftarrow{Q}_\mu(x', A) = \frac{\int \mathbb{1}_A(x) q(x, x') \mu(dx)}{\int q(u, x') \mu(du)} \quad (2.3)$$

be the *reverse kernel* associated with  $Q$  and  $\mu$ , where  $\mu$  is a probability measure on  $\mathcal{X}$ . In particular, by letting  $\mu$  be some marginal distribution of  $\mathbf{X}$  we obtain the transition kernel of  $\mathbf{X}$  when evolving in reverse time. Using (2.3), the joint smoothing distribution  $\phi_{0:n|n}$  may be expressed as

$$\phi_{0:n|n}(A) = \int \cdots \int \mathbb{1}_A(x_{0:n}) \phi_n(dx_n) \prod_{k=0}^{n-1} \overleftarrow{Q}_{\phi_k}(x_{k+1}, dx_k) \quad (2.4)$$

for  $A \in \mathcal{B}(\mathcal{X})^{\otimes(n+1)}$  [6, Corollary 3.3.8]. Here  $(\overleftarrow{Q}_{\phi_k})_{k=0}^{n-1}$  are the so-called *backward kernels* describing transitions of  $\mathbf{X}$  when evolving backwards in time and *conditionally* on the given observations. Consequently, a draw  $X_{0:n}$  from  $\phi_{0:n|n}$  may be produced by first computing recursively, using (2.2), the filter distributions  $(\phi_k)_{k=0}^n$  (the forward filtering pass), simulating  $X_n$  from  $\phi_n$ , and then simulating, recursively for  $k = n-1, n-2, \dots, 0$ ,  $X_k$  from  $\overleftarrow{Q}_{\phi_k}(X_{k+1}, \cdot)$  (the backward simulation pass). This is the mentioned FFBS algorithm.

As stressed in the introduction, joint smoothing distributions can be expressed on closed form only for a very few models. The same applies for any marginals of the same, including the filter distributions, and thus the decomposition (2.4) appears, at a first glance, to be of academic interest only. However, while there is a well-established difficulty of applying SMC methods directly to the smoothing recursion (2.1) (as resampling systematically the particle trajectories decreases rapidly the number of distinct particle

coordinates at early time steps; see Section 3.1), SMC methods may be efficiently used for approximating the filter distributions. Hence, by following [12] and replacing the filter distributions in (2.4) by particle filter estimates, we obtain an approximation of the joint smoothing distribution that is not at all effected by the degeneracy of the genealogical particle tree. This issue will be discussed further in Section 3.1.

### 3 Algorithms

#### 3.1 Particle smoothing

A particle filter approximates the filter distribution  $\phi_k$  at time  $k$  by a weighted empirical measure

$$\phi_k^N(dx) \stackrel{\text{def}}{=} \sum_{i=1}^N \frac{\omega_k^i}{\sum_{\ell=1}^N \omega_k^\ell} \delta_{\xi_k^i}(dx), \quad (3.1)$$

where  $(\xi_k^i, \omega_k^i)_{i=1}^N$  is a weighted finite sample of so-called particles (the  $\xi_k^i$ 's) with associated importance weights (the  $\omega_k^i$ 's) and  $\delta_\xi$  denotes a unit point mass at  $\xi$ . We remark that the weights  $(\omega_k^i)_{i=1}^N$  are not normalised, i.e. required to sum to unity, which motivates the self-normalisation in (3.1).

Based on an approximation as above at time  $k$ , an approximation of  $\phi_{k+1}$  can be obtained in different ways; however, two specific operations are common to all SMC algorithms: *selection*, which amounts to dropping particles that have small importance weights and duplicating particles with larger weights, and *mutation*, which amounts to randomly moving the particles in the state space  $\mathcal{X}$ . The approach we describe below is called the *auxiliary particle filter* [17].

Given the ancestor sample  $(\xi_k^i, \omega_k^i)_{i=1}^N$ , one iteration of the auxiliary particle filter involves sampling the auxiliary distribution

$$\begin{aligned} \Phi_{k+1}^N(\{i\} \times A) \\ \stackrel{\text{def}}{=} \frac{\omega_k^i \int g_{k+1}(x) Q(\xi_k^i, dx)}{\sum_{\ell=1}^N \omega_k^\ell \int g_{k+1}(x) Q(\xi_k^\ell, dx)} \left( \frac{\int \mathbb{1}_A(x) g_{k+1}(x) Q(\xi_k^i, dx)}{\int g_{k+1}(x) Q(\xi_k^i, dx)} \right) \end{aligned}$$

on the product space  $\mathcal{X} \times \{1, \dots, N\}$ , using some proposal distribution

$$\Pi_{k+1}^N(\{i\} \times A) \stackrel{\text{def}}{=} \frac{\omega_k^i \vartheta_k^i}{\sum_{\ell=1}^N \omega_k^\ell \vartheta_k^\ell} R_k(\xi_k^i, A)$$

where  $R_k$  is a proposal kernel on  $\mathcal{X}$  and  $(\vartheta_k^i)_{i=1}^N$  is a set of adjustment multiplier weights. As a motivation for this we note that  $\sum_{i=1}^N \Pi_{k+1}^N(\{i\} \times A)$  is the mixture distribution obtained by simply plugging the weighted empirical measure (3.1) into the filtering recursion (2.2); thus, by simulating a set of particle positions and indices from (3.1) and discarding the latter, a sample of particles approximating  $\phi_{k+1}$  is obtained. This procedure may then be repeated recursively as new observations become available in order to obtain weighted particle samples approximating the filter distributions at all time points. We will throughout this paper assume that the adjustment multiplier weights are generated from the ancestor sample according to  $\vartheta_k^i \stackrel{\text{def}}{=} \vartheta_k(\xi_k^i)$ , where  $\vartheta_k : \mathcal{X} \rightarrow \mathbb{R}^+$  is weight function. In addition we will assume that the proposal kernel has a transition density  $r_k : \mathcal{X}^2 \rightarrow \mathbb{R}^+$  with respect to  $\lambda$ . The latter implies that also  $\Pi_{k+1}^N$  has a density, which we denote by the same symbol, on  $\{1, \dots, N\} \times \mathcal{X}$ . In practice a draw from  $\Pi_{k+1}^N$  is produced by first drawing an index  $I = i$  with probability proportional to  $\omega_k^i \vartheta_k^i$  and then simulating a new particle location  $\xi$  from the measure  $R_k(\xi_k^I, dx)$ . Each of the draws  $(\xi_{k+1}^i, I_{k+1}^i)_{i=1}^N$  from  $\Pi_{k+1}^N$  is assigned the importance weight

$$\omega_{k+1}^i \stackrel{\text{def}}{=} \omega_{k+1}(\xi_k^{I_{k+1}^i}, \xi_{k+1}^i) \propto \frac{d\Phi_{k+1}^N}{d\Pi_{k+1}^N}(\xi_{k+1}^i, I_{k+1}^i) ,$$

where  $\omega_{k+1}(\cdot) : \mathcal{X}^2 \rightarrow \mathbb{R}^+$  is the importance weight function given by

$$\omega_{k+1}(x, x') \stackrel{\text{def}}{=} \frac{g_{k+1}(x')}{\vartheta_k(x)} \frac{q(x, x')}{r_k(x, x')} . \quad (3.2)$$

Finally, since the original target distribution is the marginal of  $\Phi_{k+1}^N$  with respect to the particle position, a weighted sample approximating the former is obtained by discarding the indices  $I_{k+1}^i$  and returning  $(\xi_{k+1}^i, \omega_{k+1}^i)_{i=1}^N$ .

The scheme is initialised by drawing  $(\xi_0^i)_{i=1}^N$  independently from some initial instrumental distribution  $\rho_0$  on  $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$  and assigning each of these initial particles the importance weight  $\omega_0^i \stackrel{\text{def}}{=} \omega_0(\xi_0^i)$  where, for  $x \in \mathcal{X}$ ,  $\omega_0(x) \stackrel{\text{def}}{=} g_0(x) d\rho/d\rho_0(x)$ .

Under suitable conditions the approximation  $\phi_k^N$  is consistent in the sense that, as  $N$  tends to infinity,

$$\phi_k^N(h) \xrightarrow{\mathbb{P}} \phi_k(h) ,$$

for all  $\phi_k$ -integrable target functions  $h$  [see 9, for some convergence results on the auxiliary particle filter]. In addition, as a by-product, an asymptotically



consistent estimate of the normalising constant  $Z_n$  can be obtained as

$$Z_n^N \stackrel{\text{def}}{=} \frac{1}{N^{n+1}} \left( \prod_{k=0}^{n-1} \sum_{\ell=1}^N \omega_k^\ell \vartheta_k^\ell \right) \sum_{\ell=1}^N \omega_n^\ell. \quad (3.3)$$

We remark that as a *particle* one may view not only the actual position  $\xi_k^i$  at time  $k$ , but also the whole trajectory  $(\xi_0^{G_0^i}, \xi_1^{G_1^i}, \dots, \xi_{k-1}^{G_{k-1}^i}, \xi_k^i)$ , where the indices  $(G_k^i)_{k=0}^{n-1}$  of the genealogical path are defined recursively backwards through  $G_{k-1}^i = I_k^{G_k^i}$  with  $G_n^i = i$ , of positions that led up to this current position. The particle filter may thus be used not only to approximate the filter distribution  $\phi_k$ , but also to approximate the joint smoothing distribution  $\phi_{0:k|k}$  by viewing the trajectory associated with  $\xi_k^i$  as a draw from this distribution. The set of all such histories is often referred to as the *genealogical tree*. The problem with this approach, in its basic form, is that for time-points  $k$  smaller than  $n$ , the particles  $(\xi_n^i)_{i=1}^N$  will tend to originate from the set small set of ancestors at time  $k$ . This problem is known as *degeneration* of the genealogical tree, and typically it happens that for  $k$  small enough, all particles alive at  $n$  originate from the same single particle at time  $k$ . The conclusion is that drawing particle trajectories ending with  $\xi_n^i$  will thus produce a poor estimate of the smoothing distribution  $\phi_{k:k|n}$  for  $k$  just a bit smaller than  $n$ , as there are in practice only a small collection of particles being sampled at that time-point. Backward sampling, to be described in the following, is a remedy to avoid this problem.

Given a sequence  $(\phi_k^N)_{k=0}^n$  of filter approximations obtained in a prefatory pass with the auxiliary particle filter, a particle approximation of  $\phi_{0:n|n}$  may, as mentioned in Section 2, be obtained by replacing each filter distribution  $\phi_k$  in (2.4) by the corresponding particle estimate  $\phi_k^N$ . This yields the estimator

$$\phi_{0:n|n}^N(A) \stackrel{\text{def}}{=} \int \cdots \int \mathbb{1}_A(x_{0:n}) \phi_n^N(dx_n) \prod_{k=0}^{n-1} \overleftarrow{Q}_{\phi_k^N}(x_{k+1}, dx_k) \quad (3.4)$$

for  $A \in \mathcal{B}(\mathcal{X})^{\otimes(n+1)}$ . The estimator (3.4) was recently analysed in [8], establishing its convergence to  $\phi_{0:n|n}$  in several probabilistic senses. By definition (2.3), each measure  $\overleftarrow{Q}_{\phi_k^N}(x, \cdot)$ ,  $x \in \mathcal{X}$ , has support at the particles  $(\xi_k^i)_{i=1}^N$  only and the weight of each support point  $\xi_k^i$  is given by  $\omega_k^i q(\xi_k^i, x) / \sum_{\ell=1}^N \omega_k^\ell q(\xi_k^\ell, x)$ . Thus the estimator  $\phi_{0:n|n}^N$  is impractical since the cardinality of its support grows exponentially with  $n$ . However, a *draw* from  $\phi_{0:n|n}^N$  is straightforwardly obtained using the following algorithm.

**Algorithm 1**

(\* particle-based FFBS \*)

1. run the particle filter to obtain  $(\xi_k^i, \omega_k^i)_{1 \leq i \leq N, 0 \leq k \leq n}$
2. simulate  $J_n \sim (\omega_n^i)_{i=1}^N$
3. set  $X_n^* \leftarrow \xi_n^{J_n}$
4. **for**  $k \leftarrow n - 1$  **to** 0
5.     **do** simulate  $J_k \sim (\omega_k^i q(\xi_k^i, X_{k+1}^*))_{i=1}^N$
6.     set  $X_k^* \leftarrow \xi_k^{J_k}$
7. set  $X_{0:n}^* \leftarrow (X_0^*, \dots, X_n^*)$
8. **return**  $X_{0:n}^*$

We note, for reasons that will be clear in the coming section, that Algorithm 1 provides, as a by-product of the forward filtering pass in Step 1, an estimate  $Z_n^N$  (given in (3.3)) of the normalising constant  $Z_n$ . Since computing the normalising constant of the probability distribution in Step 4 involves summing over  $N$  terms, the overall cost of executing Steps 2–7 (i.e. the backward simulation pass) is  $\mathcal{O}(nN)$ . As noted in [8], this cost can be reduced significantly in the case where the transition density  $q$  is bounded by some finite constant  $q_+$ , i.e.  $q(x, x') \leq q_+$  for all  $(x, x') \in \mathcal{X}^2$ , which is the case for a large class of models (e.g. all non-linear models with additive Gaussian noise). Indeed, by applying instead a standard accept-reject scheme where a candidate  $J_k^*$  is sampled from the probability distribution induced by the particle weights  $(\omega_k^i)_{i=1}^N$  (whose normalising constant is obtained as a by-product of the forward filtering pass) and accepted with probability  $q(\xi_k^{J_k^*}, X_{k+1}^*)/q_+$ , the corresponding complexity can be reduced to  $\mathcal{O}(n)$ . More specifically, [8, Proposition 1] proves that the number of simulations per index needed for obtaining, at any time step  $k$ ,  $N$  indices  $J_k$  of  $N$  conditionally independent replicates of the backward index chain tends to a constant in probability. We will apply this strategy for the implementation in Section 4.

**3.2 FFBS-based independent Metropolis-Hastings sampler**

Since the density of the smoothing distribution is known up to a normalising constant, state-space models can be perfectly cast into the framework of the Metropolis-Hastings algorithm. When applied to smoothing in state-space models, the output of the M-H algorithm is a Markov chain  $(X_{0:n}^{(\ell)})_{\ell \geq 0}$  on  $\mathcal{X}^{n+1}$  with the following dynamics. Given  $X_{0:n}^{(\ell)}$ , a candidate  $X_{0:n}^*$  for  $X_{0:n}^{(\ell+1)}$  is produced by simulation according to  $X_{0:n}^* \sim k_n(X_{0:n}^{(\ell)}, \cdot)$ , where  $k_n$  is some

proposal kernel on  $\mathcal{X}^{n+1}$ ; after this, one sets

$$X_{0:n}^{(\ell+1)} = \begin{cases} X_{0:n}^* & \text{w.pr. } \alpha_n(X_{0:n}^{(\ell)}, X_{0:n}^*) = 1 \wedge \left( \frac{\phi_{0:n|n}(X_{0:n}^*)k_n(X_{0:n}^*, X_{0:n}^{(\ell)})}{\phi_{0:n|n}(X_{0:n}^{(\ell)})k_n(X_{0:n}^{(\ell)}, X_{0:n}^*)} \right) \\ X_{0:n}^{(\ell)} & \text{otherwise.} \end{cases} \quad (3.5)$$

The initial trajectory  $X_{0:n}^{(0)}$  may be chosen arbitrarily. The M-H algorithm above admits  $\phi_{0:n|n}$  as stationary distribution, and under weak additional assumptions (such as Harris recurrence),  $(X_{0:n}^{(\ell)})_{\ell \geq 0}$  converges in distribution to  $\phi_{0:n|n}$  [see e.g. 19, for details]. In order to obtain an acceptance rate close to one, one should aim to simulate the candidates from a proposal distribution that is as close to  $\phi_{0:n|n}$  as possible. Recalling Algorithm 1, a natural strategy is thus to generate the candidate using the particle-based FFBS. Indeed, with  $\mathcal{L}^N(X_{0:n}^*)$  denoting the law of the draw  $X_{0:n}^*$  returned by Algorithm 1, [16, Theorem 1] shows that under rather weak assumptions there exists a constant  $C_n < \infty$  such that for all  $N \geq 1$ ,

$$\|\mathcal{L}^N(X_{0:n}^*) - \phi_{0:n|n}\|_{\text{TV}} \leq C_n/N$$

where  $\|\cdot\|_{\text{TV}}$  denotes total variation (distance);  $\|\mu - \nu\|_{\text{TV}} = \sup_{\|f\|_\infty \leq 1} |\mu(f) - \nu(f)|$  for probability measures  $\mu$  and  $\nu$ . Unfortunately, constructing an M-H kernel based on this proposal distribution (which is *independent* of the given  $X_{0:n}^{(\ell)}$ ) is not possible in practice since the density of  $\mathcal{L}^N(X_{0:n}^*)$  is infeasible to compute. However, the joint density of all random variables (i.e. all indices and particle locations drawn in the forward pass as well as the indices obtained in the backward pass) generating the output of the particle-based FFBS has a simple form; thus, inspired by [1], we detour this difficulty by sampling instead a well chosen auxiliary target distribution on the augmented state space of all these random variables. Interestingly, it turns out that the acceptance ratio of the resulting independent M-H sampler, which is described in Algorithm 2 below, is the same as for the standard forward-smoothing-based algorithm particle independent M-H sampler proposed in [1].

## Algorithm 2

(\* FFBS-based IM-H sampler \*)

**Input:**  $X_{0:n}^{(\ell)}$

1. run Algorithm 1 to obtain  $X_n^*$  and  $Z_n^{N,*}$

2. set  $X_{0:n}^{(\ell+1)} \leftarrow X_{0:n}^*$  with probability

$$\alpha_n(X_{0:n}^{(\ell)}, X_{0:n}^*) = 1 \wedge \frac{Z_n^{N,*}}{Z_n^N};$$

otherwise set  $X_{0:n}^{(\ell+1)} \leftarrow X_{0:n}^{(\ell)}$ .

3. **return**  $X_{0:n}^{(\ell+1)}$

In order to derive precisely the scheme above, denote by  $\boldsymbol{\xi}_k \stackrel{\text{def}}{=} (\xi_k^1, \dots, \xi_k^N)$  and  $\mathbf{I}_k \stackrel{\text{def}}{=} (I_k^1, \dots, I_k^N)$  the collection of all particles and indices generated by the particle filter at time step  $k \geq 0$ . Then the process  $(\boldsymbol{\xi}_k, \mathbf{I}_k)_{k \geq 1}$  is Markovian with joint law given by the density

$$\psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n) \stackrel{\text{def}}{=} \left( \prod_{\ell=1}^N \rho_0(\xi_0^\ell) \right) \left( \prod_{k=1}^n \prod_{\ell=1}^N \Pi_k^N(i_k^\ell, \xi_k^\ell) \right), \quad (3.6)$$

Let again  $J_k$ ,  $k = n, n-1, \dots, 0$  denote the time-reversed index Markov chain of the backward smoothing pass. The joint distribution of the particle locations  $(\boldsymbol{\xi}_k)_{k=0}^n$  and indices  $(\mathbf{I}_k)_{k=1}^n$  obtained in the forward filtering pass and the indices  $(J_k)_{k=0}^n$  of the backward smoothing pass is given by

$$\begin{aligned} k_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n) \\ \stackrel{\text{def}}{=} \psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n) \times \frac{\omega_n^{j_n}}{\sum_{\ell=1}^N \omega_n^\ell} \prod_{k=1}^n \overleftarrow{Q}_{\phi_{k-1}^N}(\xi_k^{j_k}, \xi_{k-1}^{j_{k-1}}) \\ = \psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n) \times \frac{\omega_n^{j_n}}{\sum_{\ell=1}^N \omega_n^\ell} \prod_{k=1}^n \frac{\omega_{k-1}^{j_{k-1}} q(\xi_{k-1}^{j_{k-1}}, \xi_k^{j_k})}{\sum_{\ell=1}^N \omega_{k-1}^\ell q(\xi_{k-1}^\ell, \xi_k^{j_k})}, \end{aligned} \quad (3.7)$$

where the second factor is the conditional distribution of  $(J_k)_{k=0}^n$  given the particle locations and indices obtained in the forward filtering pass. Using the density (3.6), the law of a draw produced by Algorithm 1 can be expressed as, for  $A \in \mathcal{B}(\mathcal{X})^{\otimes(n+1)}$ ,

$$\begin{aligned} \mathcal{L}^N(X_{0:n}^*)(A) &= \mathbb{E}_{k_n^N}[\mathbb{1}_A(\xi_0^{J_0}, \dots, \xi_n^{J_n})] \\ &= \mathbb{E}_{k_n^N}[\mathbb{E}_{k_n^N}[\mathbb{1}_A(\xi_0^{J_0}, \dots, \xi_n^{J_n}) | \boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{I}_1, \dots, \mathbf{I}_n]] = \mathbb{E}_{\psi_n^N}[\phi_{0:n|n}^N(A)]. \end{aligned}$$

It turns out that the distribution targeted by Algorithm 2 is given by

$$\begin{aligned} \pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n) &\stackrel{\text{def}}{=} \frac{\phi_{0:n|n}(\xi_0^{j_0}, \dots, \xi_n^{j_n})}{N^{n+1}} \\ &\times \frac{\psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n)}{\rho_0(\xi_0^{j_0}) \prod_{k=1}^n \Pi_k^N(i_k^{j_k}, \xi_k^{j_k})} \times \prod_{k=1}^n \frac{\omega_{k-1}^{i_k^{j_k}} q(\xi_{k-1}^{i_k^{j_k}}, \xi_k^{j_k})}{\sum_{\ell=1}^N \omega_{k-1}^\ell q(\xi_{k-1}^\ell, \xi_k^{j_k})}, \end{aligned} \quad (3.8)$$

and the following results, whose proofs are postponed to the appendix, are the fundamental for the construction of this algorithm.

**Theorem 1** *For any particle sample size  $N \geq 1$ , the distribution of  $(\xi_0^{J_0}, \dots, \xi_n^{J_n})$  under  $\pi_n^N$  is  $\phi_{0:n|n}$ .*

**Theorem 2** *For any  $N \geq 1$ , the update produced by Algorithm 2 is a standard M-H update with target distribution  $\pi_n^N$  and proposal distribution  $k_n^N$ .*

Impose the following (standard) boundedness condition on the particle importance and adjustment multiplier weight functions.

**A 1** *For all  $0 \leq k \leq n$ ,  $\|\omega_k\|_\infty < \infty$  and  $\|\vartheta_k\|_\infty < \infty$ .*

We now have the following result.

**Theorem 3** *Let Assumption 1 hold. Then there is a  $\kappa \in [0, 1)$  such that for all  $\ell \geq 1$  and all  $x_{0:n} \in \mathcal{X}^{n+1}$ ,*

$$\|\mathcal{L}(X_{0:n}^{(\ell)} | X_{0:n}^{(0)} = x_{0:n}) - \phi_{0:n|n}\|_{\text{TV}} \leq \kappa^\ell.$$

This result relies on the fact that if the ratio of target to proposal density, here  $Z_n^{N,*}/Z_n$ , is bounded, the independence M-H sampler converges geometrically [cf. 1, p.293].

Finally, by applying an Azuma-Hoeffding-type exponential inequality for geometrically ergodic Markov chains derived recently in [10] we obtain, as a corollary of Theorem 3, the following result describing the convergence of MCMC estimates formed by the output of Algorithm 2.

**Corollary 4** *Let Assumption 1 hold. Then for all  $N \geq 1$  there exists a constant  $c > 0$  such that for all bounded measurable functions  $h : \mathcal{X}^{n+1} \rightarrow \mathbb{R}$ , all  $m \geq 1$ , all initial trajectories  $X_{0:n}^{(0)} = x_{0:n} \in \mathcal{X}^{n+1}$ , and all  $\epsilon > 0$ ,*

$$\mathbb{P} \left( \left| \frac{1}{m} \sum_{\ell=0}^m h(X_{0:n}^{(\ell)}) - \phi_{0:n|n}(h) \right| \geq \epsilon \right) \leq c \exp \left( -\frac{m}{c} \left\{ \frac{\epsilon^2}{\|h\|_\infty^2} \wedge \frac{\epsilon}{\|h\|_\infty} \right\} \right).$$

### 3.3 Rao-Blackwellisation and multiple trajectories

The results above show that the acceptance probability for a proposed trajectory obtained by backward sampling does in fact not depend on the current or proposed trajectories themselves but only on the likelihood estimates,

computed from the set of particles and their importance weights, underlying the respective trajectories. Theorem 2 in [1] shows that the same holds true when tracing a trajectory backwards from the genealogical tree, an MCMC algorithm they referred to as the *particle independent Metropolis-Hastings* (PIMH) sampler. Therefore we can view the M-H sampler as one that proposes and possibly accepts *sets of particles* rather than trajectories, and from the current set of particles we may choose to simulate a trajectory either by sampling a final state and following its genealogy backwards, or by backward sampling.

Denoting by  $X_{0:n}^{\text{sim}}$  a trajectory simulated by either method using the current set of particles, we know that

$$\mathbb{E}[h(X_{0:n}) | y_{0:n}] = \mathbb{E}\{\mathbb{E}[h(X_{0:n}^{\text{sim}}) | (\boldsymbol{\xi}_k, \boldsymbol{\omega}_k)_{0 \leq k \leq n}] | y_{0:n}\},$$

where expectations are computed under the stationary distribution of the MCMC sampler and the notation  $(\boldsymbol{\xi}_k, \boldsymbol{\omega}_k)_{0 \leq k \leq n}$  also contains the ancestral history of each particle, if required. Therefore it also holds that

$$\mathbb{E}[h(X_{0:n}) | y_{0:n}] = \mathbb{E}\left\{\mathbb{E}\left[\left.J^{-1} \sum_{j=1}^J h(X_{0:n}^{\text{sim},j})\right| (\boldsymbol{\xi}_k, \boldsymbol{\omega}_k)_{0 \leq k \leq n}\right] \middle| y_{0:n}\right\},$$

where now  $X_{0:n}^{\text{sim},j}$  denotes one of  $J$  trajectories sampled independently.

Moreover, we can in principle remove sampling of trajectories altogether by letting  $J \rightarrow \infty$ . This is equivalent to enumerating *all* possible sampled trajectories  $x_{0:n}$ , computing the probability  $v_{x_{0:n}}$  say of that trajectory being sampled, and finally computing the weighted average  $\sum v_{x_{0:n}} h(x_{0:n})$ . When sampling from the genealogical tree this is possible to do, as there are only  $N$  different possible trajectories (ending at positions  $\xi_n^i$  for  $i = 1, \dots, N$ ). Replacing sampling by averaging in this way is known as *Rao-Blackwellisation*. Section 4.6 in [1] certainly does point this out, and it also provides convergence results for the weighted average above. For backward sampling it is generally not possible to work with all possible trajectories, as there are typically  $N^{n+1}$  of them, but for low-dimensional distributions of  $X_{0:n}$ , like that of a single  $X_k$  or a pair  $(X_k, X_{k+1})$ , Rao-Blackwellisation is feasible. It can then be obtained by iterating the normalised weights at time  $n$  backwards through the backward kernels, which is equivalent to the backwards pass of the forward-backward algorithm for HMMs. Thus we obtain the smoothing probability  $v_k^i$  say that a sampled trajectory will pass through  $\xi_k^i$  at time  $k$ , and we can compute the weighted average  $\sum_{i=1}^N v_k^i h(\xi_k^i)$ . For a pair  $(X_k, X_{k+1})$ , a similar computation is possible.

Computing smoothing probabilities obviously requires more computing time than does tracing a trajectory backwards as when sampling from the genealogical tree. However, since the tree will have low variability at time points away from the final time point  $n$ , averaging over such points will involve summing over just one or a few particles. Backward smoothing does not suffer from this problem, and hence we can expect better Rao-Blackwellisation for all time-points except for the few last ones. A compromise is however also possible, namely to simulate a number, say 5–25, trajectories using backward sampling and computing the average over those. We will now take a closer look at this approach.

Write  $(\boldsymbol{\xi}^{(r)}, \boldsymbol{\omega}^{(r)})$  for the set  $(\boldsymbol{\xi}_k, \boldsymbol{\omega}_k)_{0 \leq k \leq n}$  of particles and weights in the  $r$ -th iteration of the MCMC algorithm, and let  $X_{0:n}^{(r,j)}$ ,  $j = 1, \dots, J$ , be  $J$  trajectories obtained from this set of particles using backward sampling, simulated independently. Assume for simplicity that we wish to estimate  $\mathbb{E}[h(X_k) | y_{0:n}]$  for some  $k$ ; the discussion here generalises with only notational changes to functionals of one than one  $X$ -variable.

Running  $R$  MCMC iterations,  $(1/RJ) \sum_{r=1}^R \sum_{j=1}^J h(X_k^{(r,j)})$  is our estimate of  $\mathbb{E}[h(X_k) | y_{0:n}]$ . To express the variance of this estimate, consider

$$\begin{aligned}
\text{Var} \left( \sum_{r=1}^R \sum_{j=1}^J h(X_k^{(r,j)}) \right) &= \mathbb{E} \left[ \text{Var} \left( \sum_{r=1}^R \sum_{j=1}^J h(X_k^{(r,j)}) \middle| (\boldsymbol{\xi}^{(r)}, \boldsymbol{\omega}^{(r)})_{r=1}^R \right) \right] \\
&\quad + \text{Var} \left[ \mathbb{E} \left( \sum_{r=1}^R \sum_{j=1}^J h(X_k^{(r,j)}) \middle| (\boldsymbol{\xi}^{(r)}, \boldsymbol{\omega}^{(r)})_{r=1}^R \right) \right] \\
&= \mathbb{E} \left[ \sum_{r=1}^R J \text{Var}(h(X_k^{\text{sim}}) | (\boldsymbol{\xi}^{(r)}, \boldsymbol{\omega}^{(r)})) \right] \\
&\quad + \text{Var} \left[ \sum_{r=1}^R J \mathbb{E}(h(X_k^{\text{sim}}) | (\boldsymbol{\xi}^{(r)}, \boldsymbol{\omega}^{(r)})) \right] \\
&= RJ \left\{ \sigma^2 + JR^{-1} \text{Var} \left[ \sum_{r=1}^R \mathbb{E}(h(X_k^{\text{sim}}) | (\boldsymbol{\xi}^{(r)}, \boldsymbol{\omega}^{(r)})) \right] \right\} \\
&\approx RJ \{ \sigma^2 + J \sigma_\infty^2 \},
\end{aligned}$$

where  $X_k^{\text{sim}}$  as above denotes a generic trajectory obtained by backward sampling,  $\sigma^2 = \mathbb{E}[\text{Var}(h(X_k^{\text{sim}}) | (\boldsymbol{\xi}, \boldsymbol{\omega}))]$ , and  $\sigma_\infty^2$  is the limit as  $R \rightarrow \infty$  of the normalised variance of the sum in the second last step. This limit, the so-called *time-average variance constant* (TAVC) in terminology from [3,

Chapter IV.1], will exist if the MCMC sampler mixes not too slowly. Thus we can approximate the variance of our estimate as

$$\text{Var} \left( \frac{1}{RJ} \sum_{r=1}^R \sum_{j=1}^J h(X_k^{(r,j)}) \right) \approx \frac{1}{R} (\sigma^2/J + \sigma_\infty^2). \quad (3.9)$$

Now assume that it takes time  $\tau_{\text{PF}}$  to simulate one set of particles, and that it takes time  $\tau_{\text{BS}}$  to simulate one trajectory using backward sampling. The total computational cost for obtaining the estimate above is then  $R(\tau_{\text{PF}} + J\tau_{\text{BS}})$ . If we have a total computation time  $\tau$  available, we can minimise the right-hand side of (3.9) under the constraint that the total computation time is  $\tau$ . Treating  $J$  as a continuous variable, one finds that the optimal value of  $J$  is

$$J_{\text{opt}} = \sqrt{\frac{\sigma^2/\tau_{\text{BS}}}{\sigma_\infty^2/\tau_{\text{PF}}}}.$$

This expression is quite intuitive; if the variability of  $h(X_k^{\text{sim}})$  within a fixed set of particles tends to be large ( $\sigma^2$  is large) and sampling trajectories is quick ( $\tau_{\text{BS}}$  is small), then we should reduce variability by drawing many trajectories. Likewise we should do so if variability between sets of particles is small ( $\sigma_\infty^2$  is small) and it is time-consuming to generate new sets of particles ( $\tau_{\text{PF}}$  is large).

In practice neither of the parameters involved above are known, so they need to be estimated from data and run times. In the example below we illustrate this.

### 3.4 Including a parameter

Andrieu et al. [1, Section 4.4] devised an algorithm, referred to as the *particle marginal Metropolis-Hastings* (PMMH) update, for sampling in the case where a model parameter is included in the MCMC sampler's state space. We will now outline, briefly, that an entirely similar approach is applicable when trajectories are proposed using FFBS.

Thus there is a parameter (vector)  $\theta$  in some space  $\Theta$ , and the transition density  $q$ , the emission densities  $g_k$ , and the initial distribution  $\rho$  may all depend on  $\theta$ . To  $\theta$  belongs a prior density (with respect to some dominating measure on  $\Theta$ ), denoted by  $\pi$ . The joint posterior density of  $\theta$  and  $x_{0:n}$ , which we denote by  $\pi_n(\theta, x_{0:n})$ , is then proportional to  $\pi(\theta)\phi_{0:n|n}(x_{0:n}; \theta)$ .

The MCMC algorithm uses a proposal density  $k_\theta(\cdot|\cdot)$  say for proposing new values for  $\theta$ , and is as follows.



### Algorithm 3

(\* FFBS-based PMMH sampler \*)

**Input:**  $\theta^{(\ell)}$  and  $X_{0:n}^{(\ell)}$

1. sample  $\theta^*$  from  $k_\theta(\cdot|\theta^{(\ell)})$
2. run Algorithm 1, under the parameter  $\theta^*$ , to obtain  $X_n^*$  and  $Z_n^{N,*}$
3. set  $(\theta^{(\ell+1)}, X_{0:n}^{(\ell+1)}) \leftarrow (\theta^*, X_{0:n}^*)$  with probability

$$1 \wedge \frac{k_\theta(\theta^{(\ell)}|\theta^*)Z_n^{N,*}}{k_\theta(\theta^*|\theta^{(\ell)})Z_n^N};$$

otherwise set  $(\theta^{(\ell+1)}, X_{0:n}^{(\ell+1)}) \leftarrow (\theta^{(\ell)}, X_{0:n}^{(\ell)})$

4. **return**  $(\theta^{(\ell+1)}, X_{0:n}^{(\ell+1)})$

In the same fashion as in [1], one may show that on an enlarged MCMC state space encompassing  $\theta$ ,  $(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n)$ ,  $(\mathbf{i}_1, \dots, \mathbf{i}_n)$  and  $(j_0, \dots, j_n)$ , the proposal density of the sampler is

$$k_\theta(\theta|\theta_0)k_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n; \theta)$$

where  $\theta_0$  is the current parameter and  $k_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n; \theta)$  is as in (3.7) but with dependence on  $\theta$  included, that the density targeted by the MCMC sampler is proportional to

$$\pi(\theta)\pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n; \theta)$$

with  $\pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n; \theta)$  as in (3.8) but with dependence on  $\theta$  included, and that the marginal distribution of  $(\theta, \xi_0^{J_0}, \dots, \xi_n^{J_n})$  under this target density is the posterior  $\pi_n(\theta, x_{0:n})$  (cf. Theorem 1). Moreover, under standard assumptions on irreducibility, the sequence  $(\theta^{(\ell)}, X_{0:n}^{(\ell)})$  generated by Algorithm 3 will converge in distribution to  $\pi_n(\theta, x_{0:n})$  [cf. 1, Theorem 4.4b].

Since the acceptance probability of Algorithm 3 does again not depend on the current or proposed trajectories themselves but only on the likelihood estimates, one can just as in Section 3.3 draw multiple trajectories from the current set of particles, or average over all of them using backward smoothing, to reduce the variance of sample means that approximate posterior expectations of functionals of the latent states. Also, again the same remark applies when trajectories are sampled backwards from the genealogical tree.

## 4 Example

In this section we illustrate the methods developed above for a state-space model often referred to as the *growth model*, and which is a standard example in the particle filtering literature. The model is

$$X_k = \frac{1}{2}X_{k-1} + 25\frac{X_{k-1}}{1 + X_{k-1}^2} + 8\cos(1.2k) + V_n, \quad (4.1)$$

$$Y_k = \frac{1}{20}X_k^2 + W_k, \quad (4.2)$$

with  $X_1 \sim N(0, \sigma_0^2)$ ,  $V_k \sim \text{NID}(0, \sigma_V^2)$  and  $W_k \sim \text{NID}(0, \sigma_W^2)$ . Because of the square  $X_{k-1}^2$  in the measurement equation (4.2), the filter distributions for this model are in general bimodal.

We chose parameters  $\sigma_0^2 = 5$ ,  $\sigma_V^2 = 10$  and  $\sigma_W^2 = 1$  [an example also studied in 1], and  $N = 500$  particles. We simulated a set  $y_{1:50}$  of observations, i.e.  $n = 50$ , and then  $R = 5000$  sets of particles. We used the *bootstrap filter*, i.e. the filter with all adjustment multiplier weights  $\vartheta_k^i = 1$  and proposal kernel equal to the system dynamics; in other words,  $R_{k-1}(x, \cdot)$  was the Gaussian density with mean as in the right-hand side of (4.1) and with variance  $\sigma_W^2$ . For the bootstrap filter the importance weights  $\omega_{k+1}^i(x, x')$  simply become the emission densities  $g_{k+1}(x')$ . The number of accepted proposed sets of particles was 1515, yielding an empirical acceptance ratio  $1515/R \approx 0.30$ . We did not use a burn-in period at all, as the output showed no signs of a significant initial transient.

With the aim of estimating  $\mathbb{E}[X_k | y_{1:n}]$  for each  $k$ , we did in each sweep of the MCMC algorithm, i.e. for each current set of particles,

- (i) simulate one trajectory by tracing the genealogical tree backwards;
- (ii) compute the Rao-Blackwellised average, for each  $X_k$ , over all  $N$  backward trajectories from the genealogical tree;
- (iii) simulate  $J = 25$  trajectories using backward sampling;
- (iv) run the backward smoothing algorithm to compute a smoothed average of  $X_k$ , which is the same as Rao-Blackwellising backward sampling.

We denote these four methods by *GT*, *GTRB*, *BS* and *BSM* respectively. Thus GT is what it referred to as PIMH in [1]. Backward sampling was done using the importance sampling (IS) scheme in [8, Algorithm 1]. This scheme avoids computing all backward transition probabilities when extending a trajectory one step backwards, although we did abort IS, computed

all transition probabilities and used them to simulate the state in question after 15 failed IS proposals. The average IS acceptance rate over all sets of particles and time-points was 16%.

Sample averages over the  $R$  sets of particles, and, in the case of backward sampling, over the  $J$  simulated trajectories for each set of particles, are shown in Figure 1. Obviously all methods provide the same result, which they should, so the differences lie in the variances.

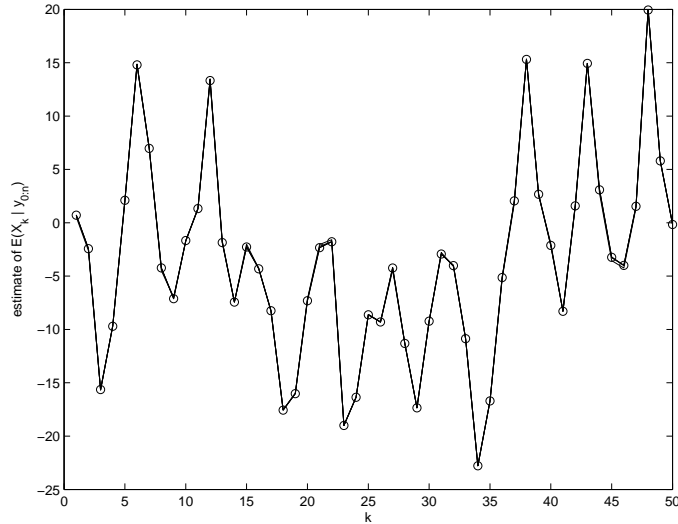


Figure 1: Estimates of  $\mathbb{E}[X_k | y_{1:n}]$  computed as sample means from the methods GT, GTRB, BS and BSM. The four curves overlap and are not distinguishable.

For BSM we have the expression  $\sigma_{\infty, \text{BSM}, k}^2 / R$  for the asymptotic variance, where  $\sigma_{\infty, \text{BSM}, k}^2$  is as in Section 3.3; observe that the expression  $\mathbb{E}[h(X_k^{\text{sim}}) | \xi^{(r)}]$ , with  $h$  as the identity function, is indeed the mean of  $X_k$  obtained with backward smoothing. Here we also include a subindex  $k$  as this variance will depend on  $k$ , and also a subindex BSM as we will require similar variances for GT and GTRB. For BS we have the asymptotic variance  $(\sigma_k^2 / J + \sigma_{\infty, \text{BSM}, k}^2) / R$  as in (3.9), where subindex  $k$  in  $\sigma_k^2$  again denotes dependence on time-index  $k$ . The asymptotic variances of GT and GTRB we write as  $\sigma_{\infty, \text{GT}, k}^2 / R$  and  $\sigma_{\infty, \text{GTRB}, k}^2 / R$  respectively, where  $\sigma_{\infty, \text{GT}, k}^2$  and  $\sigma_{\infty, \text{GTRB}, k}^2$  are TAVCs defined similarly as  $\sigma_{\infty, \text{RB}, k}^2$  but for one trajectory sampled from genealogical tree and for the weighted average over all such

trajectories respectively.

In practice neither of these variances are known, and we need to estimate them from the simulations. We estimated  $\sigma_k^2$  by first for each set of particles computing the sample variance of all  $J$  trajectories  $X_k^{\text{sim},j}$  obtained by backward sampling, and then computing the average of these sample variances over all  $R$  sets of particles. The TAVCs  $\sigma_\infty^2$  were estimated by summing up estimated autocovariances over lags  $|\ell| < \sqrt{R}$ , weighted by  $(1 - |\ell|/R)$  [cf. 5, p. 59]. Inserting these variance estimates into the expressions for asymptotic variances and taking square roots, yield standard errors for the respective estimates of  $\mathbb{E}[X_k | y_{1:n}]$ , shown in Figure 2.

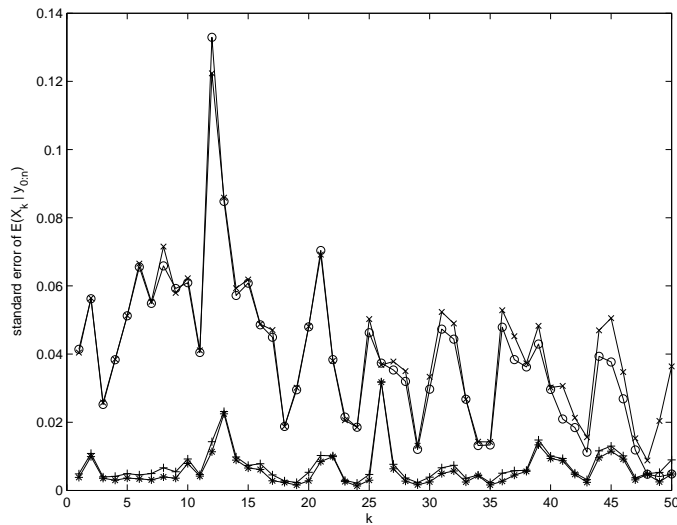


Figure 2: Standard errors of estimates of  $\mathbb{E}[X_k | y_{1:n}]$  for the methods GT ( $\times$ ), GTRB ( $\circ$ ), BS ( $+$ ) and BSM ( $*$ ).

We see that GT has standard errors larger than those of BS and BSM, which is to be expected as GT samples a single trajectory while BS samples  $J = 25$  trajectories and BSM averages over all of them. We also see that the standard errors of GT and GTRB are close to identical except towards the final time-point  $n$ . This is a result of the degeneracy of the genealogical tree as for  $k$  just a bit less than  $n$  there are only one or a few ancestors with descendants alive at time  $n$ , and then Rao-Blackwellisation (GTRB) adds little compared to just sampling (GT). For  $k \geq 43$  say GTRB however does better, and it is on par with BSM for  $k \geq 48$ ; for such late time-points

the final particles' ancestries have not coalesced and at time  $n$  GTRB and BSM are equivalent. Comparing BS and BSM we find that they have similar standard errors, and this is because the term  $\sigma_k^2/J$ , with some exceptions  $k$ , is smaller than  $\sigma_{\infty, \text{BSM}, k}^2$  for  $J = 25$ .

Comparing standard errors without comparing execution times does not provide the full picture however, and for that reason we introduce a measure of precision per computational effort, defined as inverse variance over computation time. We refer to this measure as *efficiency*, and we can estimate it using inverse squared standard errors over measured computation times. The computation time of each method was measured using the function `cputime` in `Matlab`, the software used for all simulations. Figure 3 plots these estimates. We see that BS is better than BSM, which in turn is better than GT and GTRB which perform about equally. The exception is the last few time-points for which GTRB, which is fast, does very well. The ratios of efficiencies for BS vs. GT (for all  $k$ ) ranges from 0.19 to 30.7, with 48 (out of 50) of them being larger than one and their geometric mean being 5.4. For BSM vs. GT the corresponding figures are 0.04, 11.4, 36 and 1.8 respectively.

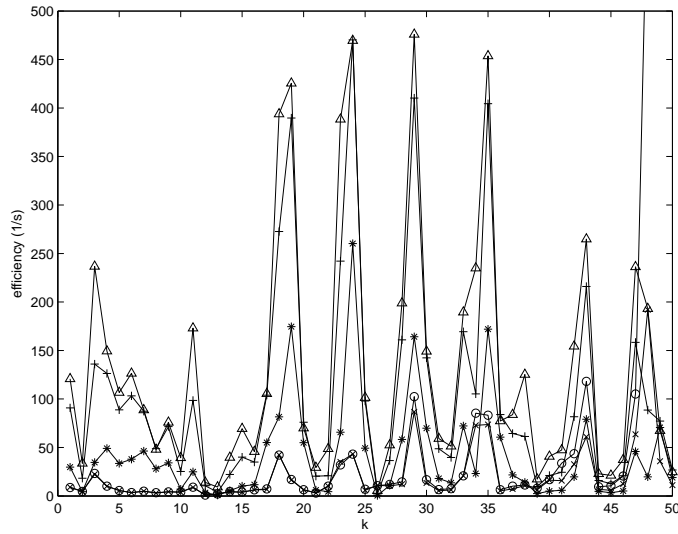


Figure 3: Estimated efficiency for estimating  $\mathbb{E}[X_k | y_{1:n}]$  for the methods GT ( $\times$ ), GTRB ( $\circ$ ), BS with  $J = 25$  trajectories ( $+$ ), BS with  $J = 7$  trajectories ( $\Delta$ ) and BSM ( $*$ ). The  $y$ -axis is truncated; GTRB reaches efficiencies about  $900 \text{ s}^{-1}$  for the final time points.

Having said that, we remark that figures like these crucially depend on software and implementation. GTRB is fast because `Matlab` does vectorised operations quickly, and we also believe that BS has a slight disadvantage from slow random number generation in `Matlab`. In addition the resolution of `cputime` appears to be 10 ms, which may not be short enough to provide accurate measures of execution times (as we measured the time of each call to functions performing the various methods).

As discussed in Section 3.3, we can choose the number  $J$  of trajectories sampled in the BS method to achieve the best variance/cost performance. This number varies quite a bit with respect to  $k$  however, with the minimum value of  $J_{\text{opt}}$  (viewed as a continuous variable) being 0.74 and the largest 18.9. As a compromise we chose the geometric mean 6.98, rounded to  $J = 7$  (the arithmetic mean is 8.12). The estimated efficiency for this  $J$  is also plotted in Figure 3, and we see that there is improvement over  $J = 25$  that is mostly marginal, but for some  $k$  notable. The ratios of efficiencies for this optimised BS vs. GT range from 0.50 to 37.6, with 49 (out of 50) of them being larger than one and their geometric mean being 7.5.

## A Proofs

### A.1 Proof of Theorem 1

To prove the statement we simply carry through the marginalisation. Thus let  $\boldsymbol{\xi}_k^{-\ell} \stackrel{\text{def}}{=} (\xi_k^i)_{1 \leq i \leq N, i \neq \ell}$  and define  $\mathbf{i}_k^{-\ell}$  analogously; the marginal of  $\pi_n^N$  with respect to  $(\xi_0^{J_0}, \dots, \xi_n^{J_n}, J_0, \dots, J_n)$  is then obtained by integrating  $\pi_n^N$  over

$(\mathbf{i}_k, \boldsymbol{\xi}_k^{-j_k})_{k=0}^n$ . We start with integrating over  $\mathbf{i}_n$  and  $\boldsymbol{\xi}_n^{-j_n}$  according to

$$\begin{aligned}
& \pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{n-1}, \xi_n^{j_n}, \mathbf{i}_1, \dots, \mathbf{i}_{n-1}, j_0, \dots, j_n) \\
&= \sum_{\mathbf{i}_n} \int \pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n, j_0, \dots, j_n) d\boldsymbol{\xi}_n^{-j_n} \\
&= \frac{\phi_{0:n|n}(\xi_0^{j_0}, \dots, \xi_n^{j_n})}{N^{n+1}} \times \left( \prod_{k=1}^{n-1} \frac{\omega_{k-1}^{j_k} q(\xi_{k-1}^{j_k}, \xi_k^{j_k})}{\sum_{\ell=1}^N \omega_{k-1}^\ell q(\xi_{k-1}^\ell, \xi_k^{j_k})} \right) \\
&\quad \times \left( \sum_{\mathbf{i}_n^{-j_n}} \int \frac{\psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n)}{\rho_0(\xi_0^{j_0}) \prod_{k=1}^n \Pi_k^N(i_k^{j_k}, \xi_k^{j_k})} d\boldsymbol{\xi}_n^{-j_n} \right) \\
&\quad \times \left( \sum_{i_n^{j_n}} \frac{\omega_{n-1}^{j_n} q(\xi_{n-1}^{j_n}, \xi_n^{j_n})}{\sum_{\ell=1}^N \omega_{n-1}^\ell q(\xi_{n-1}^\ell, \xi_n^{j_n})} \right). \quad (\text{A.1})
\end{aligned}$$

In the expression above,

$$\sum_{i_n^{j_n}} \frac{\omega_{n-1}^{j_n} q(\xi_{n-1}^{j_n}, \xi_n^{j_n})}{\sum_{\ell=1}^N \omega_{n-1}^\ell q(\xi_{n-1}^\ell, \xi_n^{j_n})} = 1 \quad (\text{A.2})$$

and

$$\begin{aligned}
& \sum_{\mathbf{i}_n^{-j_n}} \int \frac{\psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{i}_1, \dots, \mathbf{i}_n)}{\rho_0(\xi_0^{j_0}) \prod_{k=1}^n \Pi_k^N(i_k^{j_k}, \xi_k^{j_k})} d\boldsymbol{\xi}_n^{-j_n} \\
&= \frac{\psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{n-1}, \mathbf{i}_1, \dots, \mathbf{i}_{n-1})}{\rho_0(\xi_0^{j_0}) \prod_{k=1}^{n-1} \Pi_k^N(i_k^{j_k}, \xi_k^{j_k})}. \quad (\text{A.3})
\end{aligned}$$

Combining (A.1)–(A.3) yields

$$\begin{aligned}
& \pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{n-1}, \xi_n^{j_n}, \mathbf{i}_1, \dots, \mathbf{i}_{n-1}, j_0, \dots, j_n) \\
&= \frac{\phi_{0:n|n}(\xi_0^{j_0}, \dots, \xi_n^{j_n})}{N^{n+1}} \times \frac{\psi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{n-1}, \mathbf{i}_1, \dots, \mathbf{i}_{n-1})}{\rho_0(\xi_0^{j_0}) \prod_{k=1}^{n-1} \Pi_k^N(i_k^{j_k}, \xi_k^{j_k})} \\
&\quad \times \prod_{k=1}^{n-1} \frac{\omega_{k-1}^{j_k} q(\xi_{k-1}^{j_k}, \xi_k^{j_k})}{\sum_{\ell=1}^N \omega_{k-1}^\ell q(\xi_{k-1}^\ell, \xi_k^{j_k})}. \quad (\text{A.4})
\end{aligned}$$

Now, by integrating (A.4) with respect to  $(\mathbf{i}_{n-1}, \boldsymbol{\xi}_{n-1}^{-j_{n-1}})$  and repeating the same procedure for  $(\mathbf{i}_{n-2}, \boldsymbol{\xi}_{n-2}^{-j_{n-2}}), \dots, (\mathbf{i}_1, \boldsymbol{\xi}_1^{-j_1})$  and finally  $\boldsymbol{\xi}_0^{-j_0}$  we obtain

the marginal density

$$\pi_n^N(\xi_0^{j_0}, \dots, \xi_n^{j_n}, j_0, \dots, j_n) = \frac{\phi_{0:n|n}(\xi_0^{j_0}, \dots, \xi_n^{j_n})}{N^{n+1}}. \quad (\text{A.5})$$

Finally, for any rectangle  $A = A_0 \times A_1 \times \dots \times A_n$  in  $\mathcal{B}(\mathcal{X})^{\otimes(n+1)}$ ,

$$\begin{aligned} & \mathbb{P}_{\pi_n^N} \left( \xi_0^{J_0} \in A_0, \dots, \xi_n^{J_n} \in A_n \right) \\ &= \sum_{j_{0:n}} \int_{A_0} \dots \int_{A_n} \pi_n^N(\xi_0^{j_0}, \dots, \xi_n^{j_n}, j_0, \dots, j_n) d\xi_0^{j_0} \dots d\xi_n^{j_n} = \phi_{0:n|n}(A), \end{aligned}$$

implying that these measures are identical on  $\mathcal{B}(\mathcal{X})^{\otimes(n+1)}$ . We complete the proof by noting that the arguments above apply independently of the particle sample size  $N$ .

## A.2 Proof of Theorem 2

It is enough to prove that

$$\mathcal{R} \stackrel{\text{def}}{=} \frac{\pi_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{I}_1, \dots, \mathbf{I}_n, J_0, \dots, J_n)}{k_n^N(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_n, \mathbf{I}_1, \dots, \mathbf{I}_n, J_0, \dots, J_n)} = \frac{Z_n^N}{Z_n},$$

where  $Z_n^N$  is defined in (3.3). Using that

$$\Pi_k^N(I_k^{J_k}, \xi_k^{J_k}) = \frac{\omega_{k-1}^{I_k^{J_k}} \vartheta_{k-1}^{I_k^{J_k}}}{\sum_{\ell=1}^N \omega_{k-1}^\ell \vartheta_{k-1}^\ell} r_{k-1}(\xi_{k-1}^{I_k^{J_k}}, \xi_k^{J_k}),$$

we obtain

$$\mathcal{R} = \frac{\phi_{0:n|n}(\xi_0^{J_0}, \dots, \xi_n^{J_n}) \prod_{k=1}^n \{q(\xi_{k-1}^{I_k^{J_k}}, \xi_k^{J_k}) \sum_{\ell=1}^N \omega_{k-1}^\ell \vartheta_{k-1}^\ell\} \sum_{\ell=1}^N \omega_n^\ell}{N^{n+1} \rho_0(\xi_0^{J_0}) \omega_0^{J_0} \prod_{k=1}^n \{\omega_k^{J_k} q(\xi_{k-1}^{J_{k-1}}, \xi_k^{J_k}) \vartheta_{k-1}^{I_k^{J_k}} r_{k-1}(\xi_{k-1}^{I_k^{J_k}}, \xi_k^{J_k})\}}. \quad (\text{A.6})$$

Now, by the definition (3.2) of the importance weights,

$$\omega_k^{J_k} \vartheta_{k-1}^{I_k^{J_k}} r_{k-1}(\xi_{k-1}^{I_k^{J_k}}, \xi_k^{J_k}) = g_k(\xi_k^{J_k}) q(\xi_{k-1}^{I_k^{J_k}}, \xi_k^{J_k}) \quad (\text{A.7})$$

and plugging the identity (A.7) into (A.6) gives, using that, in addition,  $\rho_0(\xi_0^{J_0}) \omega_0^{J_0} = \rho(\xi_0^{J_0}) g_0(\xi_0^{J_0})$ ,

$$\mathcal{R} = \frac{\phi_{0:n|n}(\xi_0^{J_0}, \dots, \xi_n^{J_n}) \sum_{\ell=1}^N \omega_n^\ell \prod_{k=1}^n \sum_{\ell=1}^N \omega_{k-1}^\ell \vartheta_{k-1}^\ell}{N^{n+1} \rho(\xi_0^{J_0}) g_0(\xi_0^{J_0}) \prod_{k=1}^n \{g_k(\xi_k^{J_k}) q(\xi_{k-1}^{J_{k-1}}, \xi_k^{J_k})\}}.$$



Finally, we conclude the proof by noting that

$$\phi_{0:n|n}(\xi_0^{J_0}, \dots, \xi_n^{J_n})Z_n = \rho(\xi_0^{J_0})g_0(\xi_0^{J_0}) \prod_{k=1}^n \{g_k(\xi_k^{J_k})q(\xi_{k-1}^{J_{k-1}}, \xi_k^{J_k})\}.$$

## References

- [1] Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods (with discussion). *J. Roy. Statist. Soc. B* 72, 269–342.
- [2] Andrieu, C., A. Doucet, S. S. Sumeetpal, and V. B. Tadić (2004). Particle methods for change detection, system identification, and control. *Proc. IEEE* 92, 423–438.
- [3] Asmussen, S. and P. W. Glynn (2007). *Stochastic Simulation. Algorithms and Analysis*. New York: Springer.
- [4] Baum, L. E., T. P. Petrie, G. Soules, and N. Weiss (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.* 41(1), 164–171.
- [5] Brockwell, P. J. and R. A. Davis (2002). *Introduction to Time Series and Forecasting* (2nd ed.). Springer.
- [6] Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in Hidden Markov Models*. New York: Springer.
- [7] Chib, S. (1996). Calculating posterior distributions and modal estimates in Markov mixture models. *J. Econometrics* 75, 79–97.
- [8] Douc, R., A. Garivier, E. Moulines, and J. Olsson (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *Ann. Appl. Probab.* to appear.
- [9] Douc, R., E. Moulines, and J. Olsson (2009). Optimality of the auxiliary particle filter. *Probab. Math. Statist.* 29(1), 1–28.
- [10] Douc, R., E. Moulines, J. Olsson, and R. Van Handel (2011). Consistency of the maximum likelihood estimator for general hidden markov models. *The Annals of Statistics*. to appear.
- [11] Doucet, A., N. De Freitas, and N. Gordon (Eds.) (2001). *Sequential Monte Carlo Methods in Practice*. New York: Springer.

- [12] Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential Monte-Carlo sampling methods for Bayesian filtering. *Stat. Comput.* 10, 197–208.
- [13] Durbin, R. and S. J. Koopman (2001). *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press.
- [14] Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. Ph. D. thesis, University of Oxford.
- [15] Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace Electronic Syst. Mag.* 25, 53–82.
- [16] Olsson, J., T. Rydén, and S. Stjernqvist (2010). A particle-based Markov chain Monte Carlo sampler for state-space models with applications to DNA copy number data. Submitted. Also appears in S. Stjernqvist’s PhD thesis *Modelling Allelic and DNA Copy Number Variations Using Continuous-Index Hidden Markov Models*, Centre for Mathematical Sciences, Lund University, 2010.
- [17] Pitt, M. K. and N. Shephard (1999). Filtering via simulation: auxiliary particle filters. *J. Am. Statist. Assoc.* 94(446), 590–599.
- [18] Robert, C. P., G. Celeux, and J. Diebolt (1993). Bayesian estimation of hidden Markov chains: a stochastic implementation. *Statist. Probab. Lett.* 16, 77–83.
- [19] Roberts, G. O. and J. S. Rosenthal (2004). General state space Markov chains and MCMC algorithms. *Probab. Surv.* 1, 20–71.
- [20] Schön, T., F. Gustafsson, and R. Karlsson (2011). Particle filtering in practice. In D. Crisan and B. L. Rozovskii (Eds.), *Oxford Handbook of Nonlinear Filtering*. Oxford University Press. to appear.